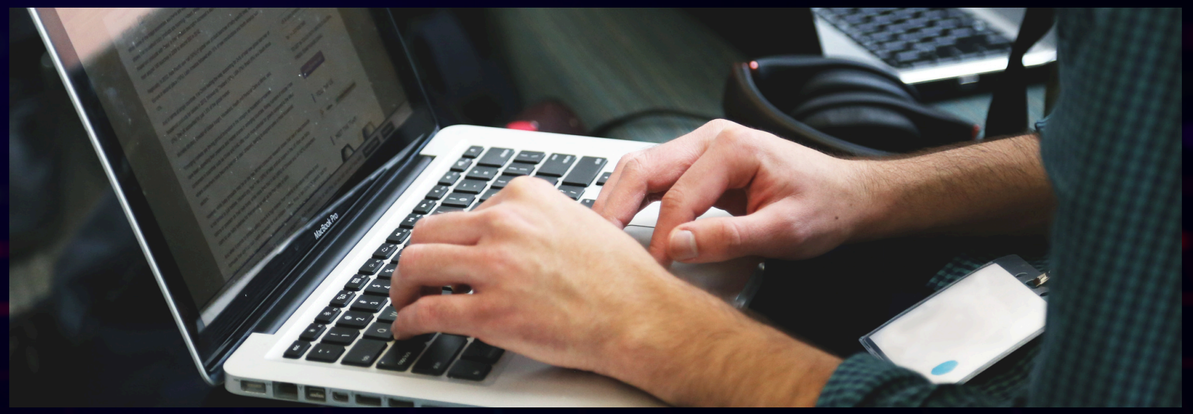
STAT 312

PRINCIPAL COMPONENT ANALYSIS

Part II

Marga, Hector, & Swapnil

OBJECTIVES



- PCA QUICK RECAP
- IMPROVEMENTS MADE
- PREPROCESSING
- DIMENSION REDUCTION
- KNN CLASSIFIER
- 3D PLOT OF NEW PCA DATASET
- SCREE PLOT & CUMULATIVE VARIANCE EXPLAINED
- VARIANCE EXPLAINED BY COMPONENT
- FEATURE CONTRIBUTIONS
- CLASSIFICATION METRICS
- CV W PCA VS W/O PCA
- BIGGEST TAKEAWAY

QUICK RECAP

We talked about how PCA works, a brief history on PCA, and paired it with Logistic Regression using Wine dataset.

IMPROVEMENTS MADE

A

Used KNN instead of Logistic Regression

B

Filled in knowledge gap by using visualizations such as scree plot to understand what's going on behind each component

C

Used CV to compare different scenarios, and find the best results.

STANDARDIZING DATA

```python
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

[6] Python

DIMENSIONALITY REDUCTION

```python
pca = PCA(n_components=3)  # Reduce to 3 components
X_train_pca = pca.fit_transform(X_train_scaled)
X_test_pca = pca.transform(X_test_scaled)
```

[7] Python

KNN CLASSIFIER

```python
knn = KNeighborsClassifier(n_neighbors=3)  #
knn.fit(X_train_pca, y_train)
```

[8] Python

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

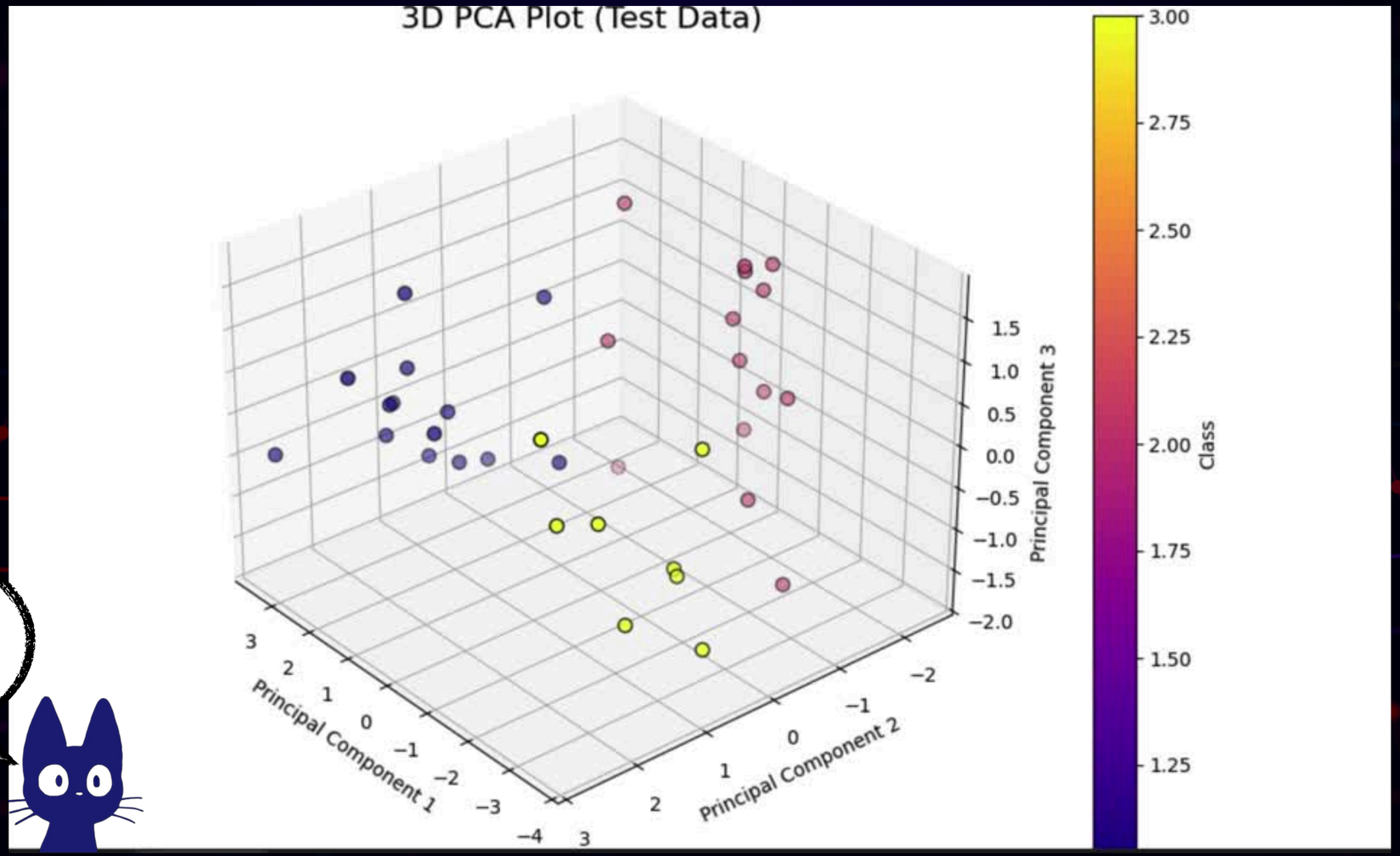
```python
y_pred = knn.predict(X_test_pca)
accuracy = accuracy_score(y_test, y_pred)
print(f"KNN Classifier Accuracy (PCA-transformed data): {accuracy:.2f}")
```

[9] Python

```
KNN Classifier Accuracy (PCA-transformed data): 1.00
```

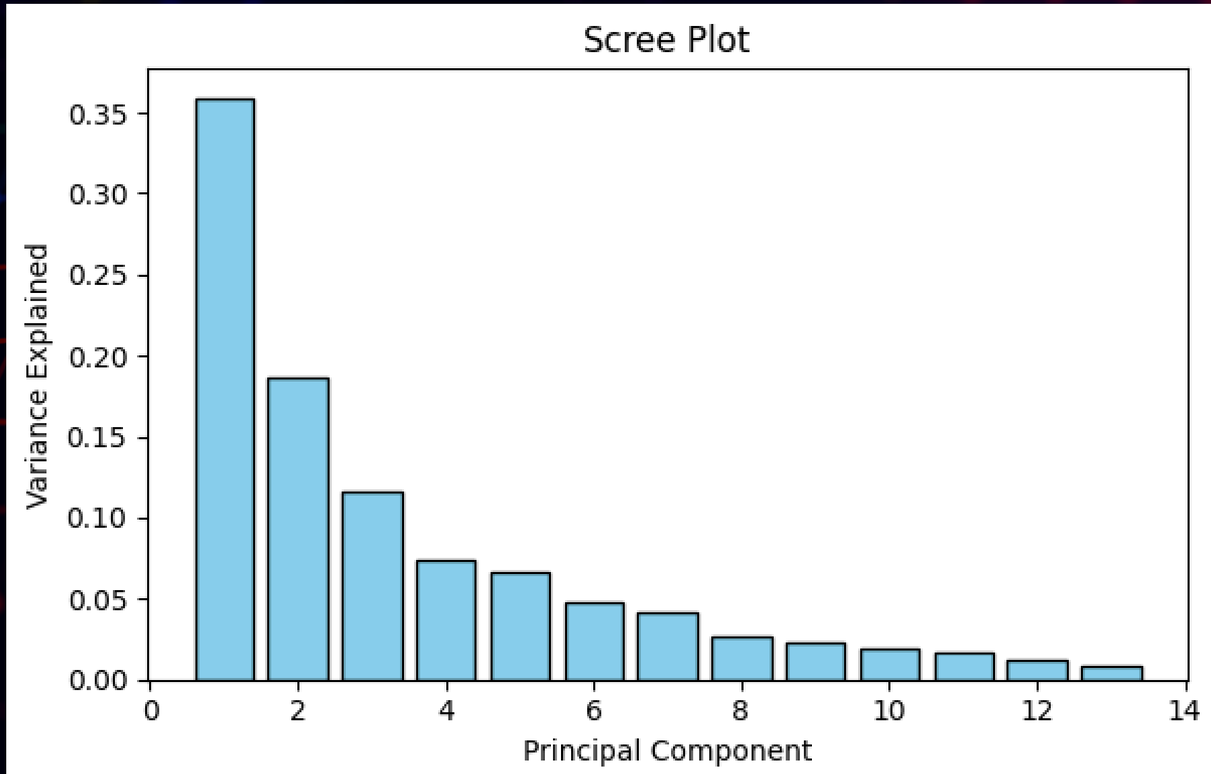
VISUALIZE PCA

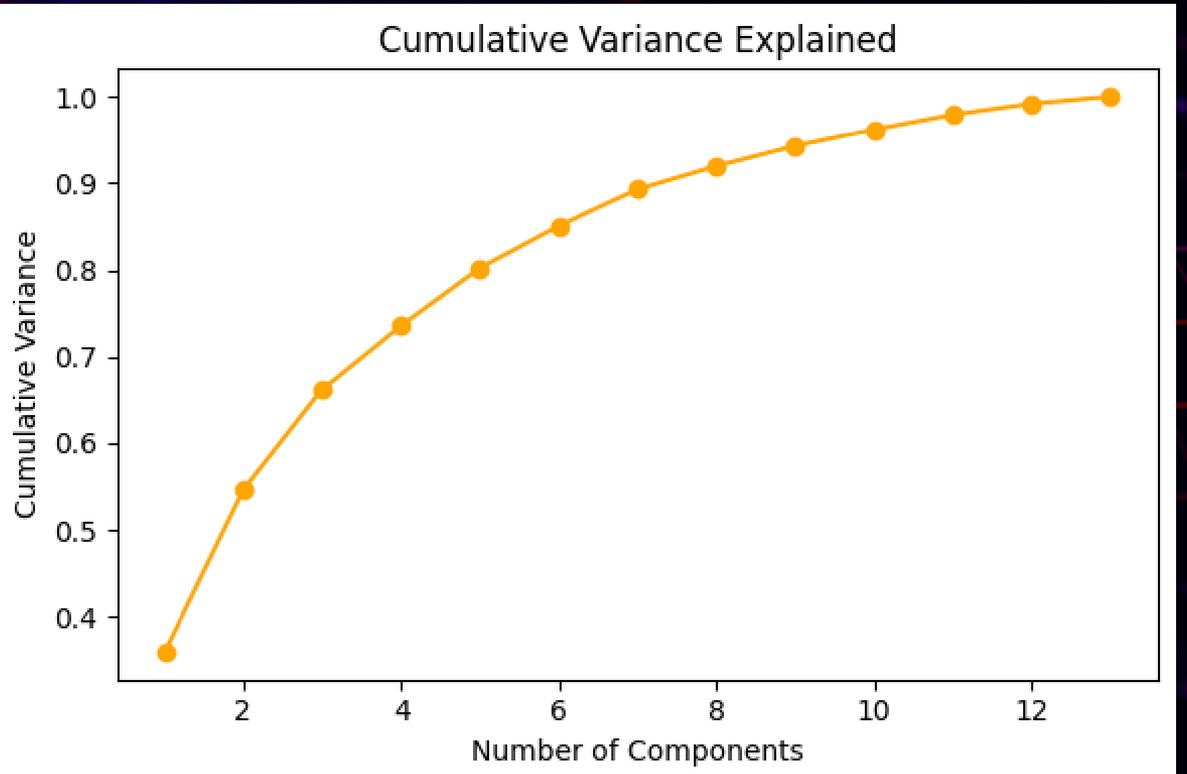
TRANSFORMED DATA



woahhh...

SCREE PLOT
CUMULATIVE VARIANCE EXPLAINED





VARIANCE EXPLAINED RATIO BY COMPONENT

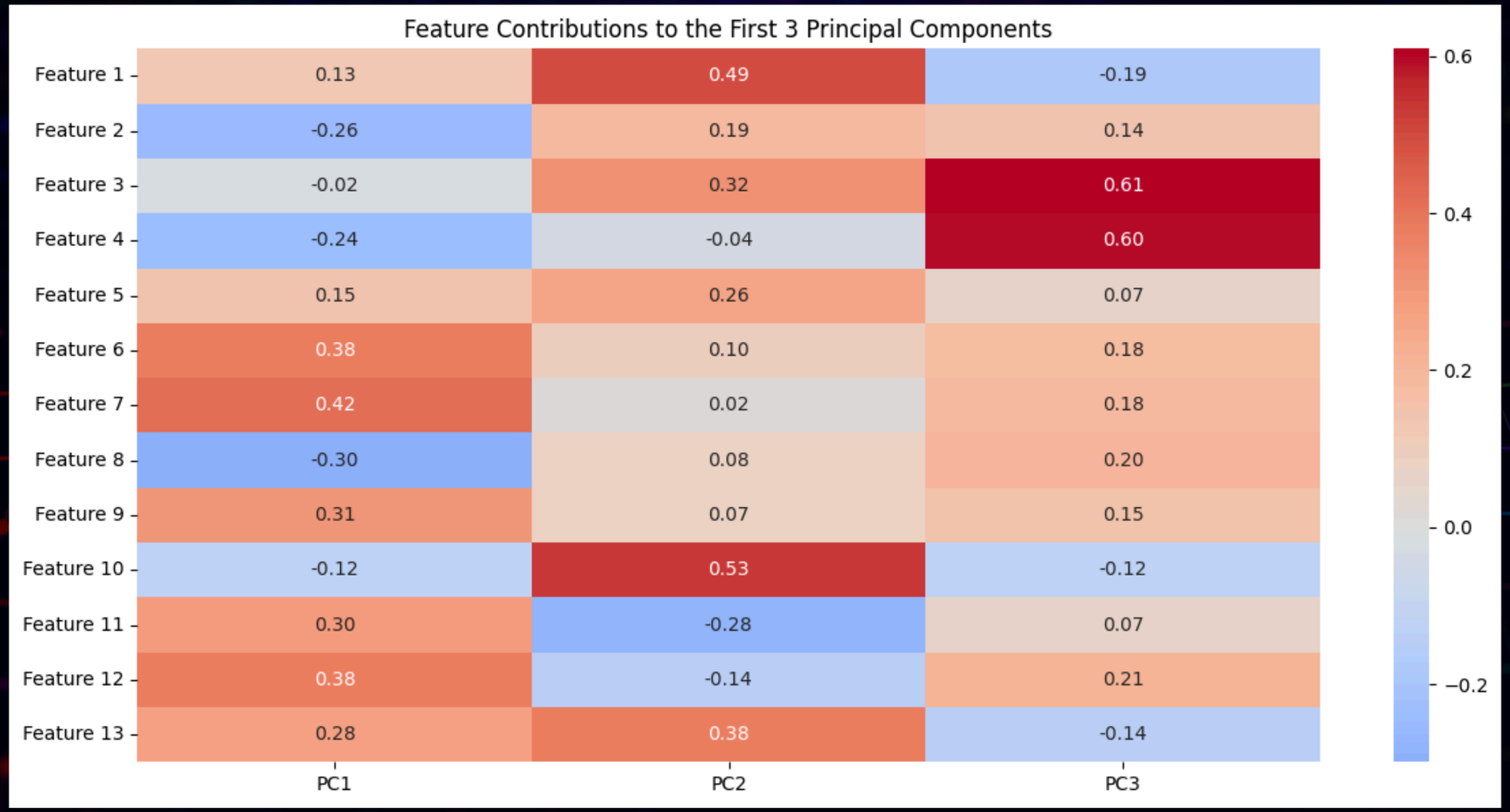
```python
# Variance explained by each component
explained_variance = pca.explained_variance_ratio_
print("Explained Variance Ratio by Component:")
for i, variance in enumerate(explained_variance, 1):
    print(f"Component {i}: {variance:.2%}")
```

[15] Python

```
Explained Variance Ratio by Component:
Component 1: 35.90%
Component 2: 18.69%
Component 3: 11.61%
Component 4: 7.37%
Component 5: 6.65%
Component 6: 4.85%
Component 7: 4.20%
Component 8: 2.68%
Component 9: 2.35%
Component 10: 1.89%
Component 11: 1.72%
Component 12: 1.26%
Component 13: 0.83%
```

If you can recall from last presentation, we had no idea how much of which feature each PC holds. But now, we do!!!

FEATURE CONTRIBUTIONS



CLASSIFICATION METRICS

```
...   Metrics for Original Data (Components: None):
                  precision    recall  f1-score   support

              1       0.93      1.00      0.97        14
              2       1.00      0.86      0.92        14
              3       0.89      1.00      0.94         8

       accuracy                           0.94        36
      macro avg       0.94      0.95      0.94        36
   weighted avg       0.95      0.94      0.94        36

   Metrics for PCA Data (Components: 2):
                  precision    recall  f1-score   support

              1       1.00      1.00      1.00        14
              2       1.00      1.00      1.00        14
              3       1.00      1.00      1.00         8

       accuracy                           1.00        36
      macro avg       1.00      1.00      1.00        36
   weighted avg       1.00      1.00      1.00        36

   Metrics for PCA Data (Components: 3):
                  precision    recall  f1-score   support
   ...
       accuracy                           1.00        36
      macro avg       1.00      1.00      1.00        36
   weighted avg       1.00      1.00      1.00        36
```

CROSS VALIDATION (NO PCA)

```
Cross-Validation Scores (No PCA): [0.96551724 0.96551724 0.96428571 0.96428571 0.92857143]
Mean Accuracy (No PCA): 0.96
```

CROSS VALIDATION WITH PCA

```
Cross-Validation Scores (With PCA): [0.93103448 0.96551724 0.96428571 0.96428571 0.89285714]
Mean Accuracy (With PCA): 0.94
```

```
Mean Accuracy with 2 PCA Components: 0.94
Mean Accuracy with 3 PCA Components: 0.94
Mean Accuracy with 5 PCA Components: 0.94
```

CONCLUSION

The wine dataset is too small for PCA.

In the process of working on the project, we learned that we have to be okay with not finding the results we want, but at least finding what doesn't work.

THANK YOU!

Thank you for exploring the essential steps and considerations in a programmer's project. Feel free to ask questions!